UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/039,289	01/04/2002	Gilbert Wolrich	10559-612001/P12851	8345

20985 7590 07/24/2007
FISH & RICHARDSON, PC
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
CHANNAVAJJALA, SRIRAMA T	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
07/24/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/039,289
Filing Date: January 04, 2002
Appellant(s): WOLRICH ET AL.

Elliott J. Mason, III
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 6/2/2006 appealing from the Office action mailed 9/28/2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after non-final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

EP0760501	Regache,Pascal	03 1997
6,438,651	Slane	08 2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. Claims 1-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Regache, Pascal [hereafter Pascal], EP 0760501 A published on 05 March 1997 in view of Slane US Patent No. 6438651.

2. As to claim 1,26,31,36, Pascal teaches a system which including 'storing in memory a plurality of a queue descriptors [col 5, line 14-21,line 22-33], Pascal specifically teaches memory having multiple memory pages that corresponds to storing in memory, further Pascal also teaches number of data items storage locations that corresponds to memory pages and data item storage locations in a circular queue, queue status that corresponds to queue descriptors,; 'each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue' [col 7, line 14-20, fig 2-3], Pascal specifically teaches circular queue maintenance of head and tail pointers, head pointer corresponds to fig 3, element 32; tail pointer corresponds to fig 3, element 30;

'fetching from memory to a cache one of either the head pointer or tail pointer of a first queue descriptor corresponding to the first queue' [col 9, line 1-15], Pascal specifically teaches checking the queue status, further data item storage locations indicated by tail pointer held in register as detailed in fig 7;

'returning to the memory from the cache portions of the first queue descriptor modified by the operation' [col 9, line 35-38];

'a count identifying a number of elements in the queue' [col 6, line 10-19, col 8, line 4-12 as claim in claim 31, claim 36], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each time a data item is written to the queue, therefore, Pascal teaches count that identifying queue size and status as detailed in col 8, line 4-12].

It is however noted that Pascal does not specifically teach 'in response to a command to perform an enqueue or dequeue operation with respect to a first queue', although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3. On the other hand, Slane specifically teaches 'in response to a command to perform an enqueue or dequeue operation with respect to a first queue' [col 3, line 52-61, fig 2], Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane's enqueue, dequeue as shown in fig 2.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory from N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

3. As to Claim 2, 15, both Pascal and Slane disclosed 'fetching the head pointer and not the tail pointer of the first queue descriptor in response to a command to perform a dequeue operation with respect to the first queue' [Pascal: col 7, line 55-9, col 8, line 1, col 9, line 11-15; Slane: col 3, line 52-55, col 4, line 47-49, col 5, line 61-63, fig 3].

4. As to Claim 3, 16, Slane disclosed 'fetching the tail pointer and not the head pointer of the first queue descriptor in response to a command to perform an enqueue operation with respect to the first queue' [col 4, line 47-51, fig 3].

5. As to Claim 4, 17, Slane disclosed 'returning to memory the head pointer and not the tail pointer of the first queue descriptor if only dequeue operations were performed on the first queue' [col 4, line 51-52, line 56-60, fig 4-6].

6. As to Claim 5, 18, Slane disclosed 'returning to memory the tail pointer and not the head pointer of the first queue descriptor if only enqueue operations were performed on the first queue while the first queue was unempty' [col 5, line 64-67, col 6, line 1-8].

7. As to Claim 6, 19, Slane disclosed 'returning to memory the head pointer and tail pointer of the first queue descriptor if an enqueue and a dequeue operation were performed on the first queue, or an enqueue operation was performed on the queue while the first queue was empty' [col 7, line 3-8].

8. As to claim 7, 33, Pascal teaches a system which including 'storing in memory a plurality of a queue descriptors [col 5, line 14-21,line 22-33], Pascal specifically teaches memory having multiple memory pages that corresponds to storing in memory, further Pascal also teaches number of data items storage locations that corresponds to memory pages and data item storage locations in a circular queue, queue status that corresponds to queue descriptors;

'each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue' [col 7, line 14-20, fig 2-3], Pascal specifically teaches circular queue maintenance of head and tail pointers, head pointer corresponds to fig 3, element 32; tail pointer corresponds to fig 3, element 30;

'determining whether a head pointer or a tail pointer of a queue descriptor that was fetched from the memory to a cache' [fig 3, col 7, line 14-16], head pointer corresponds to fig 3, head pointer, element 32; tail pointer corresponds to fig 3, element 30;

'returning one of either the head pointer or tail pointer to the memory from the cache only if that pointer had been modified' [col 9, line 35-38],

'a count identifying a number of elements in the queue' [col 6, line 10-19, col 8, line 4-12], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each time a data item is written to the queue, therefore, Pascal teaches count that identifying queue size and status as detailed in col 8, line 4-12].

It is however noted that Pascal does not specifically teach 'in response to an enqueue or dequeue operation had been modified by the enqueue or dequeue operation', although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3. On the other hand, Slane specifically teaches "in response to an enqueue or dequeue operation had been modified by the enqueue or dequeue operation' [col 3, line 52-61, fig 2], Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane's enqueue, dequeue as shown in fig 2.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory from N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

9. As to Claim 8, 20, Slane disclosed 'using valid bits in the cache to track modifications to the pointers' [col 6, line 19-25].

10. As to Claim 9, 21, Slane disclosed 'using a first valid bit to track modifications to the head pointer and second valid bit to track modifications to the tail pointer' [col 6, line 26-37].

11. As to Claim 10, 22, 29, Slane disclosed 'setting the first valid bit if a dequeue operation is performed with respect to the queue descriptor' [col 6, line 57-64], 'an enqueue operations performed with respect to the queue descriptor while the queue is empty' [col 7, line 3-12].

12. As to Claim 11, 23, Slane disclosed ‘setting the second valid bit if an enqueue operation is performed with respect to the queue descriptor’ [col 4, line 65-67, col 5, line 1-8].

13. As to Claim 12, 24, Slane disclosed ‘setting a pointer’s valid bit when the pointer is fetched from the memory to the cache’ [col 5, line 55-63].

14. As to Claim 13, 25, 30, Slane disclosed ‘returning to the memory pointers whose valid bits have been set’ [col 5, line 59-63].

15. As to claim 14,34, Pascal teaches a system which including ‘memory for storing queue descriptors [col 5, line 14-21,line 22-33], Pascal specifically teaches memory having multiple memory pages that corresponds to storing in memory, further Pascal also teaches number of data items storage locations that corresponds to memory pages and data item storage locations in a circular queue, queue status that corresponds to queue descriptors

‘a count identifying a number of elements in the queue’ [col 6, line 10-19, col 8, line 4-12 as in claim 34], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each time a data item is written to the queue;

'each of which include a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue' [col 7, line 14-20, fig 2-3], Pascal specifically teaches circular queue maintenance of head and tail pointers, head pointer corresponds to fig 3, element 32; tail pointer corresponds to fig 3, element 30;

'a cache for storing queue descriptors corresponding to up to a number of the memory's queue descriptors' [col 8, line 26-31];

'a processor configured to: fetching from memory to a cache one of either the head pointer or tail pointer of a first queue descriptor corresponding to the first queue' [col 9, line 1-15], Pascal specifically teaches checking the queue status, further data item storage locations indicated by tail pointer held in register as detailed in fig 7;

'return to the memory from the cache portions of the first queue descriptor modified by the operation' [col 9, line 35-38], It is however noted that Pascal does not specifically teach 'in response to a command to perform an enqueue or dequeue operation with respect to the particular queue descriptor', although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3. On the other hand, Slane specifically teaches 'in response to a command to perform an enqueue or dequeue operation with respect to the particular queue descriptor' [col 3, line 52-61, fig 2], Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane's enqueue, dequeue as shown in fig 2.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory from N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

16. As to claim 27, 32, 35, 37, Slane disclosed 'fetch the head pointer and not the tail pointer of the first queue descriptor in response to a command to perform a dequeu operation with respect to the first queue' [col 3, line 52-55, col 4, line 47-49, col 5, line 61-63, fig 3];

'fetching the tail pointer and not the head pointer of the first queue descriptor in response to a command to perform an enqueue operation with respect to the first queue" [col 4, line 47-51, fig 3].

17. As to Claim 28, Slane disclosed 'the head pointer and not the tail pointer of the first queue descriptor if only dequeue operations are performed on the first queue'[col 4, line 51-52, line 56-60, fig 4-6],

'the tail pointer and not the head pointer of the first queue descriptor if only enqueue operations are performed on the queue while the first queue was unempty' [col 5, line 64-67, col 6, line 1-8],

'both the head pointer and tail pointer of the first queue descriptor if both an enqueue and a dequeue are performed on the queue, or an enqueue operation was performed on the queue while the first queue was empty' [col 7, line 3-8].

(10) Response to Argument

Applicant's arguments in the appeal brief filed on 6/2/2007, with respect to the rejection of claims 1-37 have been fully considered but not persuasive, for examiner's response, see discussion below:

a) At page 7-9, applicant contends that no proper combination of these references describes or suggests the claimed invention.

At page 11-13, applicant contends that examiner has failed to provide a proper motivation to suggest the desirability of combining Pascal with Slane.

At page 13, claim 27, applicant argues that no proper combination of these references describes or suggests the claimed invention recited in claim 27.

In response to applicant's argument [a] that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Pascal is directed to data handling system with circular queue formed in paged memory, more specifically, writing data items in sequence to a circular queue formed in paged memory. from N data-item storage locations [page 3, col 3, line 3-8], it is noted that memory having multiple memory pages that corresponds to storing in memory, [col 5, line 14-21, line 22-33]. It is also noted that Pascal specifically suggests “circular queue” maintenance of “head and tail” pointers, here head pointer corresponds to fig 3, element 32; tail pointer corresponds to fig 3, element 30

Slane USPatent No. 6438651 is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, data access request is a write request to update the requested data block in the memory, the update data to the data block is written to the cache line entry including the data to update when accessing the entry in the cache line [col 2, line 7-13], further allows optimizing cache access [col 2, line 55-59]

It is however noted that Pascal does not specifically teach ‘in response to a command to perform an enqueue or dequeue operation with respect to a first queue’, although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3. On the other hand, Slane specifically teaches ‘in response to a command to perform an enqueue or dequeue operation with respect to a first queue’ [col 3, line 52-61, fig 2], Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane’s enqueue, dequeue as shown in fig 2.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory from N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

b) At page 9, claim 1, applicant argues that Pascal does not describe or even suggest that there is more than one "queue descriptor" that includes both a head pointer and a tail pointer.

c) At page 10, claim 1, Appellant contends that the examiner's interpretation of Pascal amounts to nothing more than unstated modification of Pascal......head pointer...and a tail pointer would be of no import to such a structure.

As to the argument [b-c], as best understood by the examiner, Pascal specifically suggests circular queue formed across pages for example paged memory as detailed in fig 2, further circular queue itself maintains head and tail pointers, and these pointers are maintained within a queue segment or storage locations as detailed in col 7, line 11-22, fig 3, therefore, Pascal teaches both head and tail pointers.

d) At page 10-11, claim 1, applicant argues that Pascal does not even disclose a cache, much less that a head or tail pointer is fetched from a memory to a cache.

As to the argument [d], as best understood by the examiner, firstly, Pascal is directed to paged memory structure having storage locations particularly maintains a head pointer [fig 3, element 32], and tail pointer [fig 3, element 20], secondly, Pascal also suggests not only contiguous block to the circular queue, but also "queue status" information [page 3, col 3, line 21-28]; thirdly, Pascal also suggests both "read and write" commands for example write address register for holding indicator of the address in paged memory, [col 3, line 40-45], reading data item from the storage location indicated by the read-address manager [col 3, line 56-58]. Pascal also suggest data items in the circular queue not only checks the queue status by reading operation, writing into specific storage locations, but also supports comparing values of storage location[s] as indicated by the tail pointer as detailed in col 9, line 1-15, while C-index, P-index supports status of the storage location, particularly, data items in the circular queue [col 8, line 32-45]

Examiner applies above arguments of claim 1 is representative of claims 1,7014,20-26, and 29-30.

e) At page 13, claim 27, applicant argues that neither Pascal nor Slane teaches or suggests fetching either the head pointer or tail pointer from any location to any other location in response to any command.....as recited in claim 27.

As to the above argument [e], As best understood by the examiner, Slane is directed to managing requests to a cache using "flags" to queue and dequeue in a buffer, more specifically cache line entry maintaining the data block is "accessed" to perform the data access request [see Abstract], further, Slane specifically teaches memory data structure that supports "Head and Tail" pointers particularly, circular buffer [see fig 1]. It is further noted that Slane also suggests data blocks are fetched from the memory particularly, providing the addresses for the head and tail pointers as detailed in col 4, line 47-49, col 5, line 61-63, fig 3.

Examiner applies above arguments of claim 27 is representative of claims 2-3,15-16.

f) At page 13-14, claim 28, applicant argues that "not only do none of these passages teach or suggest returning to memory "the head pointer and not the tail pointer" or "the tail pointer and not the head pointer" or "both the head pointer and tail pointer"

As to the above argument [f], as best understood by the examiner, Slane specifically teaches memory data structure that supports both "head and tail" pointers [see fig 3], further, as noted, Slane also suggests queue operation particularly both enqueue and dequeue operations performed on data structure circular buffer [col 3, line 58-61], more specifically head and tail pointers providing the addresses related to entry in the queue operations in data block [col 4, line 41-45]. Slane also suggests assigning dequeed or accessed operations from buffer particularly read from cache, further, as noted above, dequeue/enqueue operations are part of the queue operations providing addresses for "head and tail" pointers [col 7, line 3-8].

Examiner applies above arguments of claim 28 is representative of claims 4-6,17-19,and 28

g) At page 15, claim 31, applicant argues that Pascal does not suggest "fetching from the memory to a cache the count and one of either the head pointer or tail pointer"....

As to the argument [g], as best understood by the examiner, Pascal specifically teaches circular queue formed in paged memory where data or items are stored [see fig 2], further circular queue specifically teaches both "tail and head" pointers [see fig 2-3], these pointers are maintained within a queue segment i.e. to identify storage locations for both read/write operations [page 5, col 7, line 11-28], also noted that Pascal suggests checking "queue" status by reading the index value [page 6, col 9, line 1-4], comparing the index values to determine read or write the data in a storage location specified by he "tail" pointer as detailed in [page 6, col 9, line 11-15, fig 7].

It is however, noted that Pascal does not specifically disclosed 'in response to a command to perform an enqueue or dequeue operation with respect to a first queue', *although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3.*

On the other hand, examiner noted that Slane specifically teaches 'in response to a command to perform an enqueue or dequeue operation with respect to a first queue' [col 3, line 52-61, fig 2], ***Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane's enqueue, dequeue as shown in fig 2.***

Therefore,it would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory from N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

Examiner applies above arguments of claim 31-37

Therefore, applicant's remarks are deemed not to be persuasive, and claims 1-37 rejected under 35 USC 103(a) as being unpatentable over Pascal, EP 0760501 in view of Slane, US Patent No. 6438651.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Srirama Channavajjala

Conferees:

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER

Alam, Hosain
SPE 2166
July 17, 2007

Lee, Eddie
TQAS Appeal Specialist
TC2100